

VŠB - Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

Absolvování individuální odborné praxe  
Individual Professional Practise in the  
Company

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 7. května 2009

.....

podpis

## **Abstrakt**

Práce pojednává o průběhu studentské praxe ve firmě TietoEnator. Prvně představím firmu a její specializaci, dále pak pracovní zařazení studenta. Popíši jeden ze zadaných úkolů a jeho řešení. V závěru práce proberu přínos této studentské praxe, znalosti, které jsem získal, nebo znalosti, které mi naopak scházely.

## **Klíčová slova**

ASP.NET, HttpModul, regulární výraz, konfigurační sekce, TietoEnator, odborná praxe

## **Abstract**

This work deals with during students experiens in TietoEnator. From the start I introduce the Company and specialization of this Company. Next I introdukce a student's work place. I going to describe one Task and his resolution. At the End of this work I disscus the acquisition this student's experiens, knowledge which we gained, or knowledge which missed.

## **Keywords**

ASP.NET, HttpModul, regular expresion, config section, TietoEnator, profesional practise

## Obsah

1. Úvod	1
2. Firma TietoEnator	2
2.1 Představení firmy	2
2.2 Pracovní zařazení	2
3. Zvolený postup řešení zadaných úkolů	3
3.1 Zadání	3
3.2 Popis úlohy	3
3.3 Řešení úlohy	3
3.3.1 Registrace http modulu	3
3.3.2 Vytvoření vlastní konfigurační sekce	4
3.3.3 Funkce http modulu	5
4. Teoretické a praktické znalosti a dovednosti uplatněné v průběhu odborné praxe	6
5. Znalosti a dovednosti scházející během odborné praxe	6
6. Dosažené výsledky a jejich celkové zhodnocení	7

# 1.Úvod

Praxe je velice důležitá součást výuky studenta. Díky praxi má možnost uplatnit své teoretické znalosti na reálných projektech. Má možnost se seznámit se skutečnými postupy a řešeními.

V první kapitole představím firmu samotnou a pozici, kterou jsem vykonával. Dále se seznámíme s jedním z projektů, které jsem řešil v průběhu mé praxe. Popisovaný úkol je řešen pomocí ASP.NET. Dále zhodnotím množství nově nabytých zkušeností a informací. Na závěr uvedu celkové zhodnocení praxe a její přínos.

## 2.Firma TietoEnator

### 2.1.Představení firmy

TietoEnator je společnost poskytující služby v oblasti IT, výzkumu, vývoje a poradenství. Se svými 16 000 odborníky je jedním z největších poskytovatelů IT služeb v severní Evropě a je přední světovou společností ve vybraných odvětvích.

Zaměřuje se na oblasti, ve kterých má ty nejdůkladnější znalosti podnikání a potřeb jejich zákazníků. Jejich stoprocentní zaměření na potřeby zákazníka a odborné znalosti ze Skandinávských zemí je odlišují od našich konkurentů.

Na jejich hlavních trzích - severní Evropa, Německo a Rusko - se zaměřují na poskytování služeb velkým a středně velkým organizacím. Celosvětově spolupracují se svými zákazníky v oblasti telekomunikací a digitálních služeb a v lesním, ropném a plynárenském průmyslu. Úzce spolupracují s předními světovými společnostmi a organizacemi a rostou společně s nimi.

### 2.2.Zařazení studenta

Mé pracovní zařazení v této firmě bylo software specialist. Pracoval jsem s technologií .NET.

## 3. Zvolený postup řešení zadaných úkolů

### 3.1. Zadání

Cílem mého úkolu bylo vytvořit aplikaci, která bude monitorovat počet návštěv na stránky firmy. Aplikace ukládá adresy stránek, na které uživatel vstoupil, adresy vyhledávacích serverů, ze kterých přišel a slova, pomocí kterých stránky ve vyhledávači našel.

### 3.2. Popis úlohy

Veškerá data potřebná pro správný běh aplikace jsou uložena v uživatelské url adrese, se kterou na stránky firmy vstupuje.

Nejideálnější řešení je vytvořit http modul, který slouží k vyčlenění kódu sdíleného více www aplikacemi. Třída, která se bude starat o průběh celé aplikace, se nazývá SearchModule. Tato třída musí implementovat rozhraní `System.Web.IHttpModule`. To specifikuje dvě metody: `Init` a `Dispose`.

Metoda `Dispose` slouží k úklidu neřízených prostředků po ukončení aplikace. U http modulu bývá často prázdná, protože se s neřízenými prostředky nepracuje.

Pro aplikaci je důležitá metoda `Init`, která se zavolá při inicializaci modulu. V argumentu metody je předán typ `HttpApplication`, který obsahuje události, na které můžeme reagovat. Pro naši aplikaci je podstatná událost `BeginRequest`, která se spouští jako první. Kromě http modulu je nutné také vytvořit pomocnou třídu, která se bude starat o zpracovávání dat a jejich ukládání na SQL server.

Url adresy se zpracovávají pomocí regulárních výrazů. Regulární výrazy se používají pro vyhledávání a úpravu textu. Pokud uživatel neví přesnou strukturu nebo délku textu, popřípadě text může mít více variant, může napsat regulární výraz. Ten zkontroluje, zda daný text odpovídá námi definovaným variantám. Regulární výrazy jsou uloženy v konfiguračním souboru `Web.config`. Do souborů s příponou `.config` se ukládají konfigurační nastavení, která chceme používat v aplikaci. Tyto soubory jsou ve formátu XML.

### 3.3. Řešení úlohy

#### 3.3.1. Registrace http modulu

Aby aplikace o modulu věděla, je nutné ji zaregistrovat v konfiguračním souboru `Web.Config` v sekci `configuration/system.web/httpModules`

```
<httpModules>
<add name="SearchModule" type="ModuleNamespace.ModuleClass,ModuleAssembly" />
</httpModules>
```

Atribut `name` reprezentuje unikátní jméno modulu. Atribut `type` je rozdělen na dvě části. První část atributu reprezentuje úplnou cestu k modulu přes jmenné prostory, druhá část pak jeho assembly.

### 3.3.2. Vytvoření vlastní konfigurační sekce

Pro lepší přehlednost v konfiguračním souboru je výhodnější si vytvořit vlastní konfigurační sekci, ve které si nakonfigurujeme námi zvolená data. Novou sekci si pojmenujeme `searchStatistic`. Uvnitř sekce budou jednotlivé regulární výrazy, které budou zpracovávat url adresy. Tyto výrazy si může správce webu upravit dle svých požadavků a potřeb, aniž by musel zasahovat do kódu. Sekci zaregistrujeme v sekci `configuration/configsections`. S registrací je nutné také vytvořit třídu, která bude reprezentovat naši sekci v kódu.

```
<section name="searchStatistic" type="
SectionNameSpace.SectionClass,SectionAssembly"/>
```

Atribut `name` nám udává jednoznačné jméno reprezentující sekci. Atribut `type` je opět rozdělen na dvě části. První část udává úplnou cestu k reprezentující třídě. Druhá část je pak assembly dané třídy.

Sekce obsahuje kolekci konfiguračních elementů, které do sekce přidáváme příkazem `add`. Každý přidávaný element má dva parametry. Parametr `name` určuje jednoznačné jméno každého elementu v konfigurační sekci. V parametru `searchEngine` je uložen regulární výraz, který zpracovává uživatelskou URL adresu, s níž vstoupil na stránky. Hledaná slova, jež uživatel zadal do vyhledávače se v url adrese nacházejí v atributu „q“. Toto označení používá většina vyhledávacích serverů. V regulárních výrazech si můžeme jednotlivé části kódu pojmenovat vlastními názvy, které nám pak usnadní zpracovávání výsledku. V mém regulárním výrazu je pouze jedna sekce, která je pojmenována „Query“.

```
<searchStatistic>
<add name="1" searchEngine="(google\.[a-z]*.*[&amp;?])q=(?\'Query\'[^\&amp;?]*)"/>
</searchStatistic>
```

Třída reprezentující sekci v souboru `Web.config` se nazývá `SearchStatisticSection`. Aby mohl být v sekci libovolný počet regulárních výrazů, je nutné uvnitř sekce vytvořit kolekci elementů, do které budeme jednotlivé regulární výrazy přidávat. Třída `SearchStatisticSection` bude vracet instanci na tuto kolekci.

```
Public class SearchStatisticSection : ConfigurationSection
{
    [ConfigurationProperty(" ", IsRequired = true, IsDefaultCollection = true)]
    public SearchStatisticCollection Add
    {
        {
            get { return (SearchStatisticCollection)this[" "]; }
        }
    }
}
```

Pro jednotlivé elementy v kolekci je nutné vytvořit třídu, ve které nadefinujeme vlastnosti jednotlivých atributů. Třída konfiguračních elementů nese název `SearchEngineElement`.

```

Public class SearchEngineElement : ConfigurationElement
{
    [ConfigurationProperty(„searchEngine“, IsRequired = true, IsKey = false)]
    public string Regexp
    {
        get { return base[„searchEngine“] as string; }
        set { base[„searchEngine“] = value; }
    }

    [ConfigurationProperty(„name“, IsRequired = true, IsKey = true)]
    public string Name
    {
        get { return base[„name“] as string; }
        set { base[„name“] = value; }
    }
}

```

### 3.3.3. Funkce Modulu

V metodě Init si zaregistrujeme posluchače na událost BeginRequest. To znamená, že náš modul se spustí na začátku požadavku.

```
Context.BeginRequest += new EventHandler(Context_BeginRequest);
```

Uvnitř metody Context\_BeginRequest se vytvoří pomocná třída SearchModuleHelper, které jsou do konstruktoru předávány dva parametry. Pomocí HttpContext.Current.Request můžeme zjistit jak adresu stránky, na kterou uživatel vstoupil, tak i adresu stránky, ze které přišel.

```

SearchModuleHelper url = new
SearchModuleHelper(HttpContext.Current.Request.Url,
HttpContext.Current.Request.UrlReferrer);

```

Ve třídě SearchModuleHelper jsou dvě metody, které se starají o zpracování dat. První je metoda Save. Ta má za úkol ukládání již zpracovaných dat do databáze. Druhou metodou je GetQueryString. Tato metoda využívá námi nadefinované konfigurační sekce. Pomocí regulárních výrazů, nadefinovaných v konfiguračních sekcích, zpracuje uživatelskou url adresu. Pokud adresa obsahuje atribut q, uloží hodnotu atributu do vnitřní proměnné SearchModuleHelper. Poté se zavolá metoda Save a provede se uložení do databáze. V databázi se uloží adresa stránky, na kterou uživatel vstoupil, adresa stránky ze které uživatel přišel a dotazovací slovo, které je uloženo v adrese v atributu q. Pokud adresa atribut q neobsahuje, vnitřní proměnná zůstane prázdná a metoda Save se nezavolá.

Tímto modulem má vlastník webu přehled o návštěvnosti jeho stránek a také má možnost zjistit pomocí jakých slov stránky jednotliví uživatelé našli.



## **4. Teoretické a praktické znalosti a dovednosti uplatněné v průběhu odborné praxe**

Během praxe jsem uplatnil znalosti z těchto předmětů:

- Programování v C#
- Architektura technologie .NET
- Teorie zpracování dat
- Databázové a informační systémy

## **5. Znalosti a dovednosti scházející během odborné praxe**

- Velice nízká informovanost o programovém prostředí ASP.NET

Problematiku bylo nutné studovat od začátku z materiálů nalezených na internetu, nebo z knih.

## **6.Dosažené výsledky a jejich celkové zhodnocení**

Získal jsem mnoho zkušeností s programováním s pomocí ASP.NET. Zjistil jsem, jak v praxi funguje práce více lidí na jednom projektu a jak je důležitá komunikace mezi jednotlivými členy týmu. Naučil jsem se používat nástroje, které se používají pro sdílení dat mezi jednotlivými programátory.

Tato praxe byla pro mě neocenitelnou zkušeností. Měl jsem možnost sledovat a podílet se na vývoji skutečných projektů.